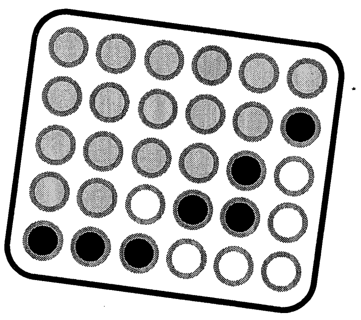
Exercise 10 – Connect Four



Connect Four is a two player logic game in which players take it in turns to drop a coloured piece of plastic into a grid, until one player has four colours matched in a row, column or diagonal.

To represent this grid on a computer, we can use ***two-dimensional arrays****. Then, each value in the array can either be an 'R' or a 'B', depending on whether the tile in that space is red or black.*

Array: A data type that can hold multiple values of the same data type.

A program that performs functions needed to play the game is shown below (and is provided electronically).

Study the code and try to understand what is happening in the program, before attempting the questions that follow.

```
def draw(grid):
    print("")
    print("1 2 3 4 5") # Print column headers
    print("| | | | |")
    print(grid[0][0], grid[0][1], grid[0][2], grid[0][3], grid[0][4], "- row 1")
    print(grid[1][0], grid[1][1], grid[1][2], grid[1][3], grid[1][4], "- row 2")
    print(grid[2][0], grid[2][1], grid[2][2], grid[2][3], grid[2][4], "- row 3")
    print(grid[3][0], grid[3][1], grid[3][2], grid[3][3], grid[3][4], "- row 4\n")

def add_piece(grid, column, row, player):
    if player == 1:
        piece = "B"
    else:
        piece = "R"
    grid[row][column] = piece
    return grid

#- MAIN PROGRAM --------------------------------------------

board = [[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0]]
won = False

draw(board)

while not won == True:

    player = 1

    print("It is player " + str(player) + "'s go.")
    c_choice = int(input("Enter the column number. "))
    r_choice = int(input("Enter the row number. "))

    board = add_piece(board, c_choice, r_choice, player)

    if player == 1:
        player = 2
    else:
        player = 1

    draw(board)

print("Player " + str(player) + " has won!")
```

NOTE: *In this program, it will not close itself unless you have completed the tasks in Section B. To manually exit the program press Ctrl+C.*

SECTION A

A 1

The first error that you might notice is that the player does not change every turn. Describe below why this happens, and then fix your program electronically.

1 1

....

....

Program updated ☐

A 2

The column and row selection also does not work correctly. Describe why this does not work below, and fix the problem on your electronic copy of the program.

1 1

....

....

Program updated ☐

A 3

Try entering a negative column, or a column greater than 5. Describe below what happens in the program when you try to enter a negative column.

1 2

....

....

Add validation to the program after lines 30 and 31 to ensure that the choice of column or row is valid. The program should repeatedly prompt the player to enter a number until it is valid.

Program updated ☐

A 4

You can currently overwrite other people's moves by choosing the same space as them.

Describe below the action you should take to stop this from happening, and update your electronic program to reflect this. *If the space is not empty, a message should be shown saying that the player has forfeited their turn.*

1 2

....

....

Program updated ☐

A 5

In the real game, you cannot put the game pieces anywhere – they either need to be on the bottom row, or on top of another existing piece.

3

Instead of asking for a row number, it makes more sense to only ask for a column, and then place the piece in the lowest empty tile on the board. Change your program so that it no longer requires the user to enter a row, and places a piece in the lowest free space on that column.

By modifying the add_piece() function, add verification to only allow a move to be made if the column has at least one empty space. If a move is invalid, you should print a suitable message and let the next player take their turn.

Program updated ☐

SECTION B

B 1

The grid in Connect Four actually uses seven columns and six rows.

Change the draw() function to allow *any* sized grid to be drawn. *The size of the grid should be calculated from the grid parameter.*

Test that it works by changing line 20 to:

```
board = [[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]]
```

You may assume that the number of rows or columns will not be greater than 9.

Program updated ☐

3

B 2

Change the program so that it asks the user to enter the size of the board (maximum 9 × 9) and then initialises the board correctly.

Program updated ☐

4

B 3

Your program currently runs forever. Describe the reason for this.

1 8

....

....

To fix this, some checking needs to be put in place to determine whether someone has won the game. Write a function check_winner() that checks whether there are four pieces in a row, *horizontally* or *vertically.*

Use this function directly after adding a piece to stop the program if there are four pieces in a row (horizontally or vertically).

For two additional marks, extend this to check for upward diagonal winning lines and, for a final two marks, extend this to check for downward diagonal winning lines.

Program updated ☐

Total: 29	5	24